Software design is an art of visualizing electricity. The moment we turn our computers on, electricity gets translated into words, pictures, and videos. We design tools, to help this translation, and those tools act as a mediators between us and the shiny world of electricity.

Most software requires user interaction, but not all. NASA drones that flew to other planets contained software programs written in LISP programming language to send and receive messages back to earth, among other things as well. Those programs do not need user interaction. On the other hand, programs that run on computers, laptops, notepads, and cellphones, they all require user interaction.

Those programs are written by artists of electricity, by software designers, who craft light and electricity into abstraction world with its own rules, understandings, means of handling and reaching things, and of-course means to make money.

Twenty years ago, software was the means to solve very tough mathematical problems, cutting significantly on time it took to solve those problems. Now, software is more applicable as a means to sell something, and as well as sharing mechanism.

Sharing and selling in software world, should have emerged together, but it did not happen this way. In the beginning, the selling and how to sell using software came up to the world, giving birth to software such as `online shops`, and `online stores`.

Each merchant and a person who has a product to sell, is very much interested from the beginning to be able to showcase and show his product to as many people and individuals as possible. The more the better. Thus more customers and more probability that someone will buy the product.

But what concerns software world, it did not happen this way. The `sharing` mechanism, or the means through which the merchant can gain exposure of his product to the maximum amount of people, emerged only twenty years later.

By `sharing`, I mean sharing software like social media sharing across the web, as well as `exposure`, which is also considered sharing.

By `exposure`, I mean exposing your product to a maximum number of people. And as an example, we can take major social networking sites, where you literally pay money for your product to get exposed to friends of friends, and to friends of friends of friends.

In computer terminology, sharing is considered a phenomenon, where two computers, residing apart from each other, can talk to one another. Moreover, it is also defined as a phenomenon, where two programs, on two separate computers, can talk to one another. Or, to the people who are familiar with object oriented approach to programming, where two objects, residing on two distant and separate computers, can talk to one another.

The evolution of computer software sharing phenomenon was greatly aided by something called `Web Services`, and Web Services in its turn was lifted up by `object oriented approach` in software, whatever programming language you use. Because before, we only understood software as structured set of instructions given to a machine to process them, one after the other. Web services, on the other hand, are already written programs, that use different protocols, to bind and join two remote objects together. HTTP and TCP are two best known protocols or `pipes`.

Windows Communication Foundation was a big hit when it was first introduced. With it, you can make a web service based on HTTP protocol, and if HHTP communication between computers is banned in some areas, it offers, Remoting, over TCP protocol. Two objects are also joined, but with Remoting Ideology and TCP pipes.

For the general stream of web, and mass media world, and maybe for the sake of simplicity, WebAPI has taken over the wheel. It is a little bit analogous to TypeScript and AngularJS, where in the beginning, there was typescript, then AngularJS was born.

Thank you,

Jinan Kordab

jinankordab.somee.com